

文章

姚鑫 · 九月 19 阅读大约需 3 分钟

第三十七章 在 UNIX®、Linux 和 macOS 上使用 IRIS (二)

第三十七章 在 UNIX®、Linux 和 macOS 上使用 IRIS (二)

在 UNIX® 上启动

IRIS 实例使用以下资源来控制启动、停止和创建新进程：

- install-dir/mgr 目录中的 iris.ids 文件。
- 共享内存。

守护进程资源锁

IRIS 使用建议文件锁定来防止同一实例在不同机器上的多次启动。使用咨询文件锁定，单个锁定文件（在本例中为 install-dir/mgr 目录中的文件 ilock）可用于独占锁定多个资源。控制进程、写入守护进程和日志守护进程分别锁定锁定文件的一个单独部分。如果 ilock 文件的这一部分已被锁定，则启动将终止。不同守护进程持有的锁称为守护进程资源锁。

文件锁由进程持有，直到进程终止。因此，如果持有任何锁，则表明某个节点上的某个守护进程正在运行。但是，它并不表示实例是否健康并正常运行。

iris.ids File

iris.ids 文件包含启动 IRIS 的节点的名称。iris.ids 文件的存在充当 ObjectScript 实用程序和客户编写的脚本的标志，指示实例是否启动并运行。该文件在启动过程中经常被忽略。但是，如果在读取 iris.ids 时发生错误，则会阻止 IRIS 启动。在以前版本的 IRIS 中，共享内存标识符也存储在 iris.ids 文件中，但现在不再如此。

启动顺序

为了更好地理解启动顺序，假设实例可以从两 (2) 个不同的节点（机器），节点 A 和节点 B 运行。iris.ids 文件对两个节点都是可见的，守护进程资源锁（对于共享文件）。然而，共享内存本身仅在创建它的节点（即启动 IRIS 的节点）上可见。

步骤 1. 检查实例状态

启动例程运行 `irisdb -cV` 以找出实例的状态。它首先尝试附加到实例的共享内存：

- 如果实例没有共享内存，则对 Daemon Resource Locks 进行测试：
 - 如果没有持有守护进程资源锁，则报告实例“关闭”。
 - 如果持有守护进程资源锁，则报告该实例正在 iris.ids 文件中指定的节点上运行。如果 iris.ids 文件不存在，则没有关于守护程序运行位置的信息。

操作：用户必须运行 `iris stop` 或 `iris force` 来停止相应节点上正在运行的实例。这将停止守护进程并删除 `iris.ids` 文件。

- 如果附加成功，则假定系统已启动并正在运行。将此状态报告给用户。启动停止。
- 如果显示错误提示无法完成启动，因为仍然附加共享内存，请等待几分钟以释放内存。如果错误仍然存在，则可能存在与 `iris.shid` 文件相关的错误，该文件在除 macOS 之外的所有 UNIX® 平台上跟踪 IRIS 的共享内存 ID。联系 全球响应中心在新选项卡 (WRC) 中打开以获取更正文件的支持。

步骤 2. 启动 RIS

运行 IRIS 启动过程。重复检查以确保另一个启动不竞争启动资源：

- 如果守护进程资源锁被持有，表明一个或多个守护进程正在这个实例的某个节点上运行，IRIS 报告这一点并退出并出现错误。启动停止。

如果 `iris.ids` 文件不存在，则运行守护程序的节点是未知的。

行动：用户必须假设在某个节点上发生了另一次启动。要确定实例在哪个节点上启动，请检查 `iris.ids` 文件。

IRIS 继续启动。

[#Caché](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC%E4%B8%89%E5%8D%81%E4%B8%83%E7%AB%A0-%E5%9C%A8-unix%C2%AE%E3%80%81linux-%E5%92%8C-macos-%E4%B8%8A%E4%BD%BF%E7%94%A8-iris%EF%BC%88%E4%BA%8C%EF%BC%89>